

## **REMARKS/ARGUMENTS**

In the Final Office Action, the Examiner has rejected claims 1, 4-11, 13-33, 35-40 42 and 43 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,804,818 (*Codella et al.*). The Examiner's rejection is fully traversed below.

Claim 1 recites: "receiving at an asynchronous proxy an asynchronous request from a first object-oriented component residing at a first server to invoke a second object-oriented component residing at a second server wherein the request has a void return type and is not associated with application-specific exceptions." In the Final Office Action, the Examiner has asserted that *Codella et al.* teaches a request having a void return type and not associated with application-specific exceptions (Final Office Action, page 4, citing Column 9, lines 28-33 of *Codella et al.*)

The pertinent section of *Codella et al.* is reproduced below for the Examiner's convenience:

Alternatively, a message bean may receive a reply as an anonymous invocation, as outlined below. This means that the method in the message proxy 104 that the message bean 102 invokes has no return type, given that the message bean 102 expects no result to be returned by the message proxy 104. [*Codella et al.*, Col. 9, lines 28-33]

It is noted that the "method in the message proxy 104 that the message bean 102 invokes has no return type. However, it is respectfully submitted that this does NOT teach that the request is not associated with application-specific exceptions. Accordingly, it is respectfully submitted that the Examiner's rejection is improper and should be withdrawn.

In addition, claim 1 further recites: "setting an exception listener on the asynchronous proxy and a scope of the second component, the exception listener being registered for the second component." It is noted that *Codella et al.* teaches that: "to differentiate a result message from an exception message sent back as a reply, the conventional 'Result' and 'Exception' labels are used respectively to tag the objects packed in the reply message" (*Codella et al.*, Col. 16, lines 46-50). It is further noted that *Codella et al.* states that: "the output port 122 associated with the method may still

define a reply-to destination. If so, then the message proxy 104 creates a MessageListener object to listen for replies and invoke the appropriate method on the message bean 102, provides the listener with the reply-to destination in the listener's corresponding output port 122, for the listener to listen to, and provides the listener with the id of the sent message for the listener to correlate replies with" (*Codella et al.*, Col. 13, lines 55-63). However, it is respectfully submitted that *Codella et al.* does NOT teach or suggest: "setting an exception listener on the asynchronous proxy and a scope of the second component, the exception listener being registered for the second component."

Furthermore, it is respectfully submitted that *Codella et al.* does NOT teach or suggest : "Providing a thread for identifying the received request and invoking the second component, wherein the thread identifies an exception listener object-oriented component for handling exceptions associated with the invocation of the second component, wherein the exception listener is registered on an asynchronous proxy, is stateless and is operable to handle a plurality of types of exceptions from a plurality of different components." It is noted that *Codella et al.* teaches: "anonymously integrating object-oriented software components and message-oriented clients wherein a first object-oriented component performs the steps of performing invocations which are serviced by one of message-oriented clients and object-oriented components and servicing the invocations which are performed by one of the message-oriented clients and the object-oriented components such that the first object-oriented component is unaware that the invocations are performed and serviced by one of the message-oriented clients and the object-oriented components" (*Codella et al.*, Abstract). However, it is respectfully submitted that *Codella et al.* does NOT teach or suggest a thread identifying an exception listener for handling exceptions associated with invocation of a second component based on a request from a first component of an object-oriented computing environment, such that the exception listener is registered on an asynchronous proxy, is stateless and operable to hand a plurality of types of exceptions from a plurality of different components.

Other independent claims recite one or more features discussed above and are therefore believed to be patentable over *Codella et al.* for at least the same reasons.

## **CONCLUSION**

Based on the foregoing, it is submitted that the claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P820). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP

/RMahboubian/  
Ramin Mahboubian  
Reg. No. 44,890

September 15, 2006

P.O. Box 70250  
Oakland, CA 94612-0250  
(650) 961-8300